

OBJECT-ORIENTED PROGRAMMING LAB 2: JAVA, HOW TO PROGRAM (CONT.)

I. Objective

In this second tutorial, you will:

- Practice with array in Java.
- Have basic knowledge about object, how to create object in Java and practice it with the sample class which defined by Java.

II. Array

Java provides a data structure, the array, which stores a fixed-size sequential collection of elements of the same type. To declare an array, we have 4 possible ways:

```
dataType[] arrayName;  
dataType arrayName[];  
dataType[] arrayName = new dataType[arraySize];  
dataType[] arrayName = {value0, value1, ..., valueK};
```

Example:

```
public class MyFirstProgram  
{  
    public static void main(String[] args)  
    {  
        int[] a = {1, 2, 3, 4, 5};  
        int sum1 = 0, sum2 = 0;  
        for (int i = 0; i < a.length; i++)  
        {  
            sum1 = sum1 + a[i];  
        }  
        System.out.println("sum1 = " + sum1);  
        for (int x : a)  
        {  
            sum2 = sum2 + x;  
        }  
        System.out.println("sum2 = " + sum2);  
    }  
}
```

We have a special for loop in the above sample called *enhanced for* loop. The *enhanced for* loop is mainly used to traverse a collection of elements including arrays. The syntax is as follows.

Example:

```
public class MyFirstProgram
{
    public static void main(String[] args)
    {
        int[] a = {1, 3, 5, 7, 9};
        for (int x : a)
        {
            System.out.println(x);
        }
    }
}
```

III. Classes and objects in Java

Java is an Object-Oriented Programming (OOP) language. Everything you work in Java is through classes and objects. In this lab, we just learn how to create an object from the available class in Java, we will learn about this topic more carefully in Lab 4.

Integer is a class, which is defined in the **java.lang.Integer** package. This class wraps a value of the primitive type *int* in an object. You can read more about this class in [here](#).

Object is an instance of a Class.

Example:

```
class Test{
    public static void main(String[] args){
        Integer num = new Integer(3);

        Integer num1 = new Integer(5);

        Integer x = num;

        Integer y;

        System.out.println(num);
        System.out.println(num1);
        System.out.println(x);
        //System.out.println(y);
    }
}
```

Output:

```
3
5
3
```

With the above example, the variables: *num*, *num1*, *x*, *y* created from the Integer class. In other words, the variables: *num*, *num1*, *x*, *y* are the pointers to the objects. But with variable *y*, it hasn't been

initialized, so its reference points to **null**. With **null**, you can't do anything and it may cause **NullPointerException** when you try to use it.

The format of code when you want to create an object from the class is:

```
ClassName instanceName = new ClassConstructor([parameter1, parameter2, ...])
```

Let observe another example:

```
class Test1{
    public static void main(String[] args){
        Integer x = new Integer(3);
        Integer y = new Integer(3);

        System.out.println(x == y);
        System.out.println(x.equals(y));
    }
}
```

Output:

```
false
true
```

With objects, we don't use operator "==" to compare values, because this operator will compare the addresses and there are the pointers so they have different address. Integer supports the method called **equals** to compare the value of two Integer objects.

Next, we will learn how to invoke the **non-static method** and **static method** from a class. To invoke a static method, you do not need to create an object, instead you can invoke directly from class name. Contrarily, with non-static method, you need to create an object to invoke it.

Example:

```
class Test2{
    public static void main(String[] args){
        Integer x = new Integer(3);

        int y = x.intValue(); //intValue() is a non-static method of Integer class

        int z = Integer.sum(2,3); //sum() is a static method of Integer class

        System.out.println(y);
        System.out.println(z);
    }
}
```

Output:

```
3
5
```

IV. Exercises

1. Write a function *public static int findMax(int arr[])* to find the maximum value of an array.
2. Write a function to find the minimum value of an array.
3. Write a function to sum all even numbers of an array.
4. Write a function to count how many specific element in an array.
5. Write a function to count how many prime number in an array.
6. Write a function *public static int find(int arr[], int k)* to find the index of an element *k* in an array, if the element doesn't exist in an array return -1. (the first element index is 0)
7. Write a function *public static void square(int arr[])* to square all elements of an array.
8. Write a function *public static Integer findMax(Integer []arr)* to find the maximum value of an Integer object array.
9. *Write a function *public static int[] divisibleNumbers(int arr[], int k)* to find the elements which divisible by *k* in an array. (Ex: *a = [1,2,3,4,5,6,7]* with *k = 2* → *[2,4,6]*)
10. *Write a function to find the third largest element in an array.